Rapid Free-Space Mapping From a Single Omnidirectional Camera

Robert Lukierski[1], Stefan Leutenegger[1] and Andrew J. Davison[1].

***Abstract*— Low-cost robots such as floor cleaners generally rely on limited perception and simple algorithms, but some new models now have enough sensing capability and computation power to enable Simultaneous Localisation And Mapping (SLAM) and intelligent guided navigation. In particular, computer vision is now a serious option in low cost robotics, though its use to date has been limited to feature-based mapping for localisation. Dense environment perception such as free space finding has required additional specialised sensors, adding expense and complexity.**

Here we show that a robot with a single passive omnidirectional camera can perform rapid global free-space reasoning within typical rooms. Upon entering a new room, the robot makes a circular movement to capture a closely-spaced omni image sequence with disparity in all horizontal directions. A feature-based visual SLAM procedure obtains accurate poses for these frames before passing them to a dense matching step, 3D semi-dense reconstruction and visibility reasoning. The result is turned into a 2D occupancy map, which can be improved and extended if necessary through further movement.

This rapid, passive technique can capture high quality free space information which gives a robot a global understanding of the space around it. We present results in several scenes, including quantitative comparison with laser-based mapping.

I. INTRODUCTION

Even in the current era of many sensing options for mobile robots (LADAR, structured light or time-of-flight depth cameras, etc.), passive vision remains a highly attractive choice as the primary outward-looking sense in many applications. This is for practical reasons including cost, size, power requirements and resolution as well as the intuitive appeal of sensing the world in a human-like way and the huge long-term potential of vision for cognitive scene understanding.

In particular, omnidirectional optics enable a practical single camera setup to provide a view of the whole of a robot's immediate surroundings at once. It is well understood that as wide a field of view as possible is advantageous for localisation and SLAM, as illustrated for instance in the choice of omni-vision in commercial products such as the recently announced Dyson 360 Eye robot vacuum cleaner. However, there is much untapped potential in the video a robot with an omnidirectional camera can capture. In particular, visual robots have lacked a human's ability to glance around during a brief exploration of a new space in order to quickly get a global idea of its shape and key features. Current commercial products focus on tracking only. In this paper we give an omni-equippied robot the ability to rapidly understand the global free space within a room, with the goal of enabling intelligent high level planning and semantic understanding of spaces.

The strengths of omnidirectional vision are instant wide coverage and ease of correspondence during extended movement; while its weaknesses are low angular resolution and hard to calibrate projection characteristics. In this paper we address all steps of a global omnidirectional free space mapping algorithm, and our contributions are in various details as well as the impact of the whole system.

In our approach, we control a robot to perform short circular motions to capture sequences of closely-spaced frames with disparity in all horizontal directions. A feature-based matching and bundle adjustment procedure provides accurate estimates of the pose of each image. These are then used to construct an omnidirectional photoconsistency cost volume based on typically 100–160 frames. The cost volume is used to generate an omnidirectional depth map which can be transformed into a dense 3D vertex map. The key problem in attempting dense passive reconstruction indoors is that many rooms have textureless areas, and therefore the omnidirectional depth map and corresponding dense 3D geometry estimates typically have large areas where depth in unreliable, even when regularisation is applied. We therefore estimate depth standard deviation from the cost volume data and threshold to extract only the semi-dense high quality information. For use in indoor navigation, this 3D estimation is followed by reduction to two dimensions and visibility reasoning to estimate the occupancy of cells in a 2D grid.

We present results which show that in many small rooms a single circular scan, taking only 10–20 seconds, is enough to reliably find many metres squared of free space. In larger rooms, the robot makes several circular scans in sequence, moving to a new viewpoint in-between—whereby additional parts of the room are revealed since occluding obstacles are being rounded. The free space information obtained from all of these scans is then merged into one global map.

We believe that this could be a highly practical technique for a modern vision-equipped robot to get a rapid understanding of the global shape of a new room it enters. In our results we compare the free space information discovered by omnidirectional vision against that from a laser range-finder undergoing the same motions. Passive vision cannot compete with LADAR in all cases, but we argue that in low-cost robotics it is much more practical, and also has advantages in terms of capturing genuine 3D information.

II. RELATED WORK

Reconstructing 2D free-space maps using laser range-finders has been a standard robotics capability for many years

[1]Robert Lukierski, Stefan Leutenegger and Andrew J. Davison are with the Dyson Robotics Laboratory, Department of Computing, Imperial College London, London, UK {r.lukierski12, s.leutenegger, a.davison}@imperial.ac.uk

(e.g. [1]). Using occupancy grids as a free space map representation was introduced in [2] and remains still very popular to this day. Mainstream depth cameras are now beginning to make the equivalent in 3D commonplace (e.g. [3], [4], [5]). With passive RGB cameras, recovering dense depth and free space information is much more challenging, but there has been good progress in the vast computer vision literature on multi-view stereo (MVS), and recently even in real-time with methods like DTAM [6].

The non-standard projection geometry of omnidirectional cameras means that building a 3D vision system involves more complication than with standard lenses, and often much published omnidirectional work has concentrated on modelling and calibration (e.g. [7], [8], [**?**]) or estimating vanishing points (e.g. [**?**]). Recently there has been a resurgence of interest in unconventional geometry computer vision, most often high resolution panoramic images, stitched from multiple cameras. For example, Cabral *et al.* [**?**] presented a sophisticated panoramic image analysis which produces not only free space mapping but full floor plans, and our free space inference method takes inspiration from this. Nevertheless their interest was rather different from mobile robotics, with high resolution cameras, algorithms very far from real-time performance and the goal of aesthetically pleasing indoor models rather than navigable free-space.

Some work more closely related to this paper is described in [9], [10], where the authors also built a complex omnidirectional robotic vision system, but they relied on sparse EKF SLAM and thus, for dependable free space estimation, had to employ a laser scanner sensor.

Other related papers are [**?**], [**?**], where the authors estimate free space information, but from an omnidirectional stereo pair rather than a single moving camera. An omnidirectional MVS approach was presented in [11] but the quality of results is rather poor.

The authors are aware of more recent approaches to tracking [12] and mapping [13], but unfortunately these algorithms rely heavily on the perspective camera model, linear epipolar lines and currently do not work with generic, non-classical, cameras.

III. METHOD

Our system consists of a number of connected processing stages which will be explained in the following subsections. The complexity of the system is substantial, mostly due to the usage of a non-classical camera for which few off-the-shelf tools or datasets are available.

A. Camera Calibration

First, we give details on the omnidirectional camera model we use and our custom calibration method which obtains extremely low reprojection error by capturing images of a checkerboard displayed on a TFT LCD monitor.

1) Model: In our system we employ the Geyer & Barreto ([7], [8]) catadioptric camera model, which largely resembles the pinhole model with the addition of an extra parameter determining the curvature of the mirror. The model has the following parameters summarised as $\mathbf{V}$:

- $\mathbf{V}_1 = \begin{bmatrix}\varphi & \theta & \psi & t_x & t_y & t_z\end{bmatrix}$: extrinsic parameters which form a rotation matrix $\mathbf{R}_{WC}(\varphi,\theta,\psi)$ and translation vector $\mathbf{t}_W = (t_x, t_y, t_z)^\top$,
- $\mathbf{V}_2 = \begin{bmatrix}\epsilon\end{bmatrix}$: mirror shape parameter,
- $\mathbf{V}_3 = \begin{bmatrix}k_1 & k_2 & \gamma_1 & \gamma_2\end{bmatrix}$: radial and tangential distortion coefficients,
- $\mathbf{V}_4 = \begin{bmatrix}s & f_1 & f_2 & u_0 & v_0\end{bmatrix}$: pinhole camera intrinsics.

First a point in the camera coordinate frame $\mathbf{p}_C$ is projected onto the unit sphere:

$$\mathbf{p}_S = \frac{\mathbf{p}_C}{\|\mathbf{p}_C\|} =: (x, y, z)^\top . \tag{1}$$

The next step is to perform the perspective projection:

$$\mathbf{u}_u = \left(\frac{x}{z+\epsilon}, \frac{y}{z+\epsilon}\right)^\top =: (u_u, v_u)^\top , \tag{2}$$

where $\mathbf{u}_u$ is the undistorted image plane coordinate.

The $\mathbf{V}_2$ parameter alters the projection, as it depends on the mirror shape and defines therefore the camera type, e.g. $\epsilon = 0$ is a classical perspective camera, $\epsilon = 1$ is a spherical mirror catadioptric camera.

From here the model follows the classical pinhole perspective camera, with radial and tangential distortions:

$$\begin{aligned}\mathbf{u}_d &= \mathbf{u}_u + \mathbf{d}(\mathbf{u}_u, \mathbf{V}_3),\\ \mathbf{d}(\mathbf{u}_d, \mathbf{V}_3) &= \begin{bmatrix} u_u(k_1\rho_u^2 + k_2\rho_u^4) + 2\gamma_1 u_u v_u + \gamma_2(\rho_u^2 + 2u_u^2) \\ v_u(k_1\rho_u^2 + k_2\rho_u^4) + 2\gamma_2 u_u v_u + \gamma_1(\rho_u^2 + 2v_u^2)\end{bmatrix},\\ \rho_u &= \sqrt{u_u^2 + v_u^2},\end{aligned} \tag{3}$$

transforming undistorted image plane location $\mathbf{u}_u$ into its distorted counterpart $\mathbf{u}_d$. This is followed by the multiplication of the pinhole camera intrinsic matrix:

$$\boldsymbol{u} = \mathbf{K}\left(\mathbf{V}_4\right)\boldsymbol{u}_d = \begin{bmatrix} f_1 & f_1 s & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1\end{bmatrix}\boldsymbol{u}_d, \tag{4}$$

to compute the final pixel location $\boldsymbol{u}$ from $\boldsymbol{u}_d$, where the italic symbols denote homogeneous representations. We denote the overall projection as $\mathbf{u} = \mathbf{h}(\mathbf{p}_C, \mathbf{V})$.

2) Calibration: Camera calibration was performed with a chessboard pattern displayed on a LCD monitor, with advantages over a typical printout due to precision (pixel pitch), flatness and high contrast. Ten images were taken by moving the camera around to cover the entire field of view. In each image the rough location of the pattern is provided by the user, unwrapped to a rectangular patch and corners are detected. We employ Levenberg-Marquardt optimisation to minimise the following energy function with respect to the camera model parameters $\mathbf{V}$:

$$F(\mathbf{V}) = \frac{1}{2}\sum_{i=1}^{m}\left\|\mathbf{h}\left(\mathbf{R}_{WC}^\top \mathbf{p}_{Wi} - \mathbf{R}_{WC}^\top \mathbf{t}_W, \mathbf{V}\right) - \tilde{\mathbf{u}}_i\right\|_2^2 , \tag{5}$$

where m is the number of point correspondences, $\mathbf{p}_{Wi}$ is the location of a corner on the chequerboard pattern in Euclidean world coordinates and $\tilde{\mathbf{u}}_i$ is the corresponding image measurement of the same point on the image. By using this optimisation (with auto-differentiation) we routinely achieve average reprojection error around $0.1[pix]$ or below.

3) Unwrapping the image into a spherical panorama: A raw omnidirectional image makes subsequent steps like point feature matching and dense stereo complicated, so we perform a Look-Up-Table (LUT) unwrapping to a spherically mapped panoramic image for subsequent processing. This involves sub-pixel interpolation, but has the advantages that image dimensions fit well with GPU memory and that each pixel represents a uniform area on a surface of the sphere.

The final projection model for the spherical panorama, $\mathbf{h}_s$ is as follows. We use the notation of a 3D-point in the camera coordinate frame, $\mathbf{P}_C = (x_C, y_C, z_C)^\top$ and spherical image coordinates $\mathbf{u}_s = (u_s, v_s)^\top$:

$$\begin{aligned}\mathbf{u}_s &= \mathbf{h}_s(\mathbf{p}_c, w, h, a_m, a_r) \\ &= \left(w - (\arctan \frac{y_C}{x_C})\frac{w}{2\pi}, (\arccos\left(\frac{z_C}{\|\mathbf{P}_C\|_2}\right) - a_m)\frac{h}{a_r}\right)^\top ,\end{aligned} \tag{6}$$

where w and h denote the panorama image width and height in pixels, and a_m, a_r are the minimal vertical viewing angle and range respectively. These are estimated from the original camera model at the image boundaries. The pseudo-inverse of the camera model transforms input pixel coordinate $\mathbf{u}_s$ back to a Euclidean normal vector:

$$\begin{aligned}\varphi &= \pi - \frac{u}{w}2\pi \\ \theta &= a_m + \frac{v}{h}a_r \\ \mathbf{n} &= \mathbf{h}_s^+(\mathbf{u}, w, h, a_m, a_r) \\ &= (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)\end{aligned} \tag{7}$$

B. Accurate Feature-Based Motion Estimation

Our experimental robot has odometry which is not synchronized with image capture and for dense stereo reconstruction we need highly accurate camera pose estimates so we use our own keypoint-based omnidirectional structure from motion and bundle adjustment implementation to globally register all of the image frames in each experiment (even those in large rooms with multiple scan circles). Odometry factor terms are included to define overall scale.

In order to provide a good guess for the optimiser, we use robot's odometry as the initialisation for the camera $\mathbf{SE}(3)$ poses (the robot base to camera transform $\mathbf{T}_{RC}$ can be regarded as identity in our setup, see also Figure 1).

1) Features: Features are detected with the FAST corner detector [14] and described with the SIFT descriptor [15]. Note that while we did not optimise this particular part of our pipeline, this particular choice does not limit the generality of the method described in the following.

2) Feature matching: Our feature-based motion estimation runs iteratively and with each new frame matches newly detected features against the current landmark map or, for completely new features, initialize a new landmark in the map. Features are matched in an inner loop against the current landmark map, based on reprojection error in the image plane and SIFT descriptor distance. These are putative matches and with them the preliminary bundle adjustment (described below) is run. Putative match is rejected if its reprojection error is too large. This selection and rejection is repeated twice and then the final bundle adjustment is performed. When generating new landmarks it is important to have relatively uniform coverage of the viewing angles. The image is divided into 16 patches and we select keypoints in such a way that in each patch we retain at least 5 features (including already matched) - an approach also called bucketing. Each new keypoint has to be at least 10 pixels away from all the others to ensure a uniform distribution of high quality features. New landmarks are initialized at $7.5[m]$ and later bundle adjusted to the correct depth.

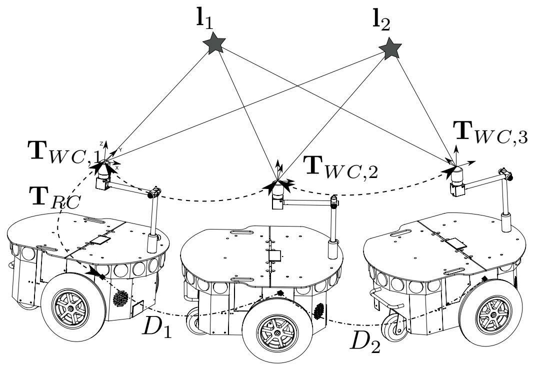


Fig. 1: Geometry of feature-based motion estimation, showing both visual and odometry constraints.

3) Bundle Adjustment: Bundle Adjustment uses non-linear optimisation to jointly estimate the locations of the landmarks (Figure 1 $\mathbf{l}_1$ and $\mathbf{l}_2$) and camera poses ($\mathbf{T}_{WC}$). The method is widely regarded a "Gold Standard" in sparse structure from motion, and offers higher accuracy per computational effort when compared to EKF filtering as described in [16]. Furthermore, we include odometry data in order to obtain scale. Thus in addition to classical factors involving cameras and landmarks we add pose constraints (D_1 and D_2 in Figure 1) that constrain distances between camera poses.

The objective is to minimise the following energy function and find accurate estimates for camera poses $\mathbf{T}_{WC,j} = [\mathbf{R}_j, \mathbf{t}_j]$ and landmarks $\mathbf{l}_i$ positions:

$$\min_{\mathbf{T}_j, \mathbf{l}_i} (E_v + E_d) . \tag{8}$$

It is composed of two parts. The visual part

$$E_v = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij} \left\| \begin{bmatrix} \frac{1}{\sigma_v} & 0 \\ 0 & \frac{1}{\sigma_v} \end{bmatrix} \left(\mathbf{h}_s(\mathbf{R}_j^\top \mathbf{l}_i - \mathbf{R}_j^\top \mathbf{t}_j) - \tilde{\mathbf{u}}_{ij}\right) \right\|_H , \tag{9}$$

imposes constraints between camera poses $\mathbf{T}_j$, landmark poses $\mathbf{l}_i$ and their respective matched keypoint detections $\tilde{\mathbf{u}}_{ij}$. The variable b_{ij} is a binary control that reflects matched features (i.e. feature i seen on frame j or not). This results in a error metric in the pixel space and is then multiplied by the precision matrix as a means of weighting between visual and odometry constraints before being normalised with Huber norm (marked $\|\|_H$). The standard deviations σ_u and σ_v in the precision matrix are set to $0.5[pix]$. The second term

$$E_d = \sum_{j=1}^{m-1} \left\| \frac{1}{\sigma_d} \left(\|\mathbf{t}_j - \mathbf{t}_{j+1}\|_2 - \|\mathbf{o}_j - \mathbf{o}_{j+1}\| \right) \right\|_C , \quad (10)$$

represents the constraints from the odometry so the final result will keep the original scale of the problem. The Euclidean distance between each two consecutive camera poses $\mathbf{T}_j$ and $\mathbf{T}_{j+1}$ is compared against the travelled distance as by the robot's odometry readings $\mathbf{o}_j$ and $\mathbf{o}_{j+1}$, which is then weighed and robustified with Cauchy loss function (marked $\|\|_C$). The standard deviation for this type of constraint σ_d is proportional to the square root of distance travelled and equal to $10[mm]$ per $200[mm]$ travelled.

The solution is obtained by the iterative Levenberg-Marquardt algorithm [17] with the Jacobians calculated by means of auto-differentiation.

C. Semi-Dense Depth Reconstruction

Now, with an accurate pose estimate for each frame, we calculate a dense omnidirectional cost volume around a reference frame, in which we seek to find the depth values for all pixels that minimise brightness discrepancies with all other images. Such methods often apply additional regularisation cost terms, in order to obtain a smooth and fully dense depth map despite the fact that some image regions may be badly conditioned for depth estimation due to the lack of texture. In the context of free space mapping, however, we had rather discard these regions where depth is essentially invented by the regularisation. We therefore resort to a semi-dense approach that only considers depth estimates of high confidence when fusing into the occupancy map.

Our 3D reconstruction method, similarly to [6] and other MVS techniques a using large numbers of images, relies on the use of a "cost volume" which is a volumetric representation where each voxel accumulates squared photometric error between images (robustified by the Huber norm). The cost volume element C_r value from reference image $\mathbf{I}_r$, for pixel $\mathbf{u}$ and depth d over the set of images $\mathcal{I}(r)$ is defined as:

$$C_r(\mathbf{u}, d) = \frac{1}{c} \sum_{m \in \mathcal{I}(r)} \|\rho_r(\mathbf{I}_m, \mathbf{u}, d)\|_H , \quad (11)$$

with c being the number of successful reprojections. The two view photometric error is defined as:

$$\mathbf{T}_{WC,m}^{-1} \mathbf{T}_{WC,r} = \begin{bmatrix} \mathbf{R}_{mr} & \mathbf{t}_m \\ \mathbf{0} & 1 \end{bmatrix},$$
$$\rho_r(\mathbf{I}_m, \mathbf{u}, d) = \mathbf{I}_r(\mathbf{u}) - \mathbf{I}_m\left(\mathbf{h}_s\left(\mathbf{R}_{mr} d \mathbf{h}_s^{+}(\mathbf{u}) + \mathbf{t}_m\right)\right). \quad (12)$$

Each row of photometric errors at pixel $\mathbf{u}$ can be then searched to find the minimal photometric error and therefore its depth. Alas, such depth estimates, even when incorporating a subpixel interpolation step, tend to be very noisy.

The method described in [6] employs a weighted Huber-ROF TV-L1 regulariser. This method performs very well in scenarios for which this algorithm was designed for, i.e. highly textured workspace scale areas. Unfortunately, for room scale data with significant textureless areas this leads to poor performance or requires parameter settings that lead to oversmoothing.

As a substantial proportion of the 3D reconstructed area has poor depth estimates we need a way to decide which measurements are trustworthy. Our method relies on estimating depth standard deviation, as illustrated in Figure 2. We show two extreme cases; (A): a low texture area in the middle of a wall, and (B): a high texture area, with contrasting energy responses in their respective cost volume depth sampling. By fitting a parabola to the minimum we can estimate the standard deviation of the depth estimate. In addition, fitting a parabola provides subpixel depth resolution. In Figure 2 parabolae for (A) and (B) have vastly different a parameters which determine the standard deviation in the inverse depth domain: $\sigma_\varepsilon(\mathbf{u}) = \frac{1}{\sqrt{2a}}$. This is then converted into standard deviation in the depth domain as follows:

$$\sigma_d(\mathbf{u}) = \frac{\sigma_\varepsilon(\mathbf{u})}{d(\mathbf{u})^2}, \quad (13)$$

where $d(\mathbf{u})$ is the subpixel depth estimate for pixel $\mathbf{u}$.

Thresholding in the depth standard deviation domain is superior to other heuristics, as demonstrated in Section IV-A. The resultant depth measurements are not now fully dense, but much more reliable for free space inference.

In our implementation the depths are represented in inverse form and the depth range is sampled into 64 bins.

D. Visibility Reasoning and Occupancy Map Estimation

With a semi-dense depth map we can now infer the amount of free space area around the reference frame camera pose. Here having spherical panoramic unwrapping is very convenient, as each column of the depth map represents a different viewing angle around the 360° field of view. Thus, for each column we need to find the closest valid depth measurement. Since we are trying to find drivable free space for a small robot, we start by looking for depth measurements in a column below the horizon row (to deal with the case where there might be free space under a table or other overhanging furniture). If no valid measurements are found then we examine the rest of the column (above the horizon), on the assumption that some vertical walls might be blank at camera height but have useful texture higher up. If no depth estimates survive standard deviation thresholding in an entire column then we assume that for this particular viewing direction the free space boundary is at ε_{max} (usually 500mm), a fail safe measure. This area is not truly measured but the safe passage of the width of the robot plus some

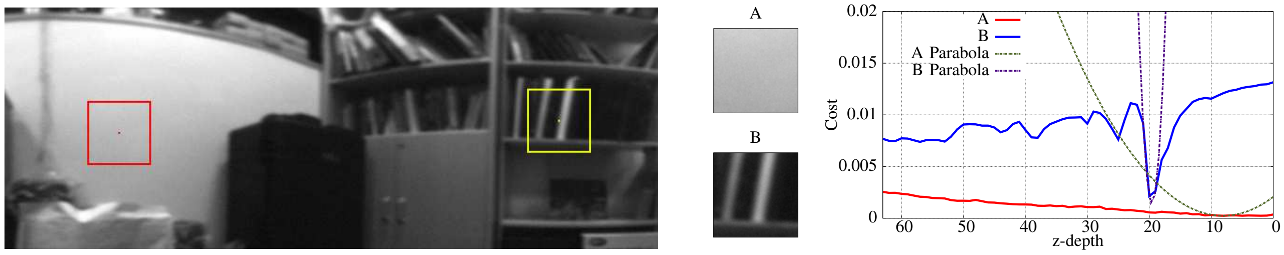


Fig. 2: Depth standard deviation estimation. Patch (A) has low texture and therefore poor depth estimate quality, with patch (B) the opposite. Fitting a parabola provides us with subpixel depth estimate as well as depth standard deviation.

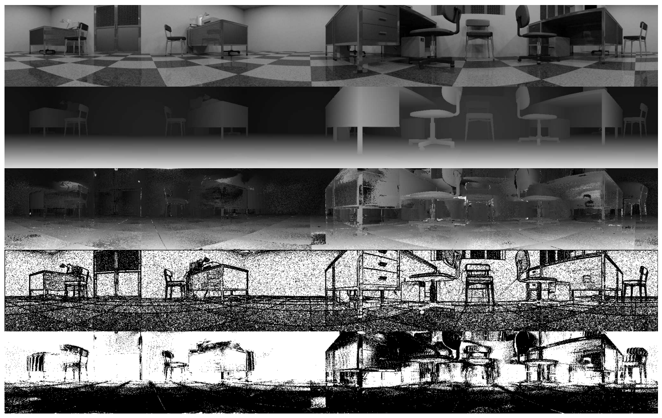


Fig. 3: Synthetic dataset, from top to bottom: reference frame, ground truth depth map, estimated depth map, image gradient thresholded to produce 50% coverage, depth standard deviation thresholded to produce 50% coverage.

margin usually provided by local sensors means that we can remove it from our occupancy maps.

For columns where a closest depth estimate is successfully found then this is used as a boundary point for the free space region and from the reference frame pose to this point a line of sight marks underlying occupancy grid cells as free space by checking line to cell intersection and increasing a free space counter of the intersected cell. Integrating hundreds of free space boundary points from a single depth map and later integrating multiple depth maps, spaced across the room and processed in the same manner, generates a global free space map in the occupancy grid. We use an occupancy map cell size of $100mm$ in all experiments.

IV. RESULTS

We present results which first evaluate our semi-dense depth map reconstruction technique using synthetic omnidirectional images we have generated using ray tracing from a modelled scene with ground truth depth. Then we move onto real experiments which test our whole system in three different office-like rooms which have significant low-texture regions similar to the target application domain of low-cost robotics in domestic environments. Our experimental setup consists of a Pioneer 3DX mobile robot platform, Point Grey Flea3 camera, Sony RPU-C2512 low-profile omnidirectional lens, SICK LMS-511 laser scanner and Nvidia CUDA capable laptop.

The laser scanner is not a part of our method, but provides precise ground truth free space measurements for comparison with our vision algorithm. The laser scanner's pose with respect to the camera was estimated using the factor graph technique of [18] to align incremental motion estimates from vision with those from laser scan matching.

For visual comparison and to help the reader understand the experimental context, we have also used a 3D RGBD dense SLAM system [19] to reconstruct each experimental room, and present screenshots from this alongside our free space maps from omnidirectional vision.

A. Synthetic Data

We have performed experiments using synthetic omnidirectional image data to investigate the performance of omnidirectional dense and semi-dense reconstruction using circular motions as in our real experiments. We have generated photorealistic data using the open source PoVRay ray-tracing software by re-rendering the data of Handa *et al.* [20] using an omnidirectional camera model, and a circular camera trajectory with 30 evenly sampled poses. We construct depth estimates from a multi-view stereo cost volume generated using all 30 images and ground truth camera poses.

The results are shown in Figure 3, where we see a comparison and between the ground truth depth map (second row) and that recovered from our omnidirectional reconstruction (third). The depth maps correspond well in areas of high texture, and as expected less well in blanker areas. We have tried two types of threshold for deciding which of these depth measurements to keep for free space mappling. In the lower two rows of Figure 3 we show the result of thresholding to retain 50% of the image area based either on a simple image gradient threshold (fourth row) or a depth standard deviation threshold as explained in Figure 2 (bottom).

Image gradient thresholding produces much more noisy output and also leaves horizontal edges where there is no real disparity for depth estimation. The depth standard deviation thresholding, on the other hand, favours better estimated areas that are closer (with finer depth sampling in the cost volume) and exhibit vertical edges.

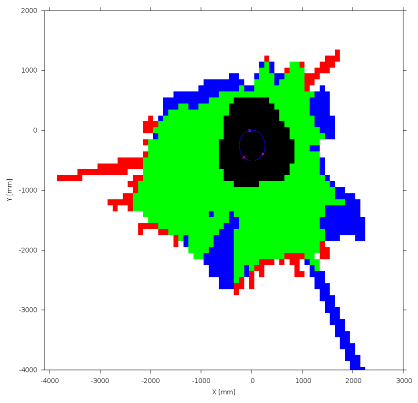
One final thing to point out is that these synthetic images include a textured floor which is observed by the camera and reconstructed. In our later real experiments, due both to the camera's limited down-angle and the lack of texture of real floors we only reconstruct room elements above the floor.

B. *Single-Circle Mapping in Small Rooms*

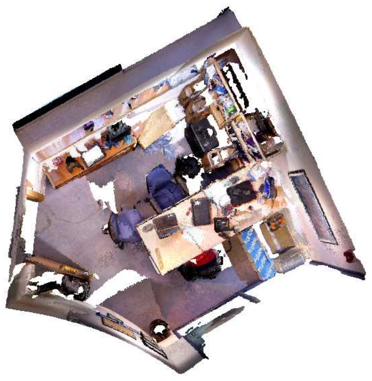
We now move to experiments using our real robot platform. In small rooms, we have tested the performance of omnidirectional depth map reconstruction and free space mapping based on a single circular motion.

During every circular motion we estimate 3 depth maps from equally spaced views. From 126 frames in the circle it takes roughly $\mathbf{6.9}[s]$, which for our application can be considered rapid.

1) Cluttered Office: This is a dataset in an office room, with the elementary motion of a 0.5m diameter circle. The robot captured 126 frames regularly spaced around the circle which were bundle adjusted to estimate accurate poses. From 3 reference poses the depth and depth standard deviation estimation procedures (Section III-C) were performed to create 3 dense depth maps with their respective standard deviation maps; Figure 4 shows one depth map result.



(a) "Cluttered Office": occupancy grid from one circular motion and three reference images, comparing our vision-based estimate with LADAR mapping. Green: vision and laser agree; blue: found by laser but not by vision; red: found by vision but not by laser; black: area close to robot known to be empty.



(b) "Cluttered Office": top-down view of the room for comparison (screenshot from an RGBD 3D SLAM system).

Fig. 5

After thresholding the depth standard deviation maps and estimating the free space boundary from the semi-dense map, each boundary was integrated into the occupancy grid shown in Figure 5a (to be compared with the visualisation in 5b). We highlight the green area, where the free space recovered from omnidirectional vision agrees with LADAR. The single circle motion, without any exploration and taking only a few seconds, uncovers over 80% of the room's floor area.

2) Empty Office: The second example we present is from an empty and newly painted office. This room is interesting because it clearly demonstrates the weaknesses of a passive stereo method in challenging, highly untextured scenes. As before, the robot drove in a 0.5m diameter circle and 3 depth map and free space measurements were estimated. Each one contributed to the occupancy grid, as seen in Figure 6.

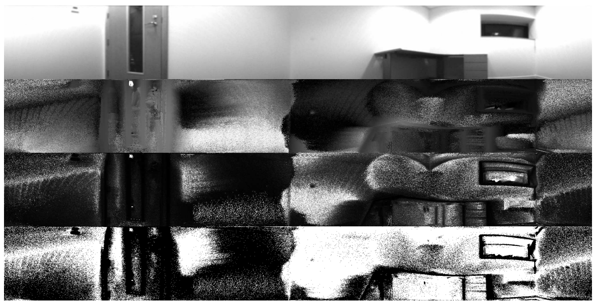


Fig. 6: Dataset "Empty Office", from top to bottom: reference frame, estimated depth Map, estimated depth standard deviation map, depth threshold mask.

As before, we draw attention to the green area in Figure 7a, which is the free space area mapped by omnidirectional vision which agrees with ladar. In this scenario the comparison between ground truth laser sensor measurements and our vision system are more meaningful, as there is almost no furniture and both sensors observe similar obstacles. However, due to the low texture conditions there are larger blue areas where meaningful depth measurements have not been made. Still, even in such unfavourable conditions for the vision system we are able to uncover 50% of the free space in the room with a rapid circular motion.

C. *Incremental Mapping in a Large Office Environment*

This dataset is of a much larger scale than the ones described in the previous subsections. It illustrates how a full free space map of a room can be built incrementally. The robot moves in a $1[m]$ diameter circle, performs all the steps of the method (Section III, i.e. estimates three depth maps and integrates free space estimates into a global occupancy grid) three times, it then traverses a short distance to another location and makes another circle. The whole trajectory is globally bundle adjusted for globally consistent poses. This means that the free space information recovered at each circle location can be fused incrementally to eventually cover almost the entire room. This is repeated 4 times across the room and each and every step uncovers more of the free space area, finally reaching almost the entire room, as can be seen in Figure 8f.

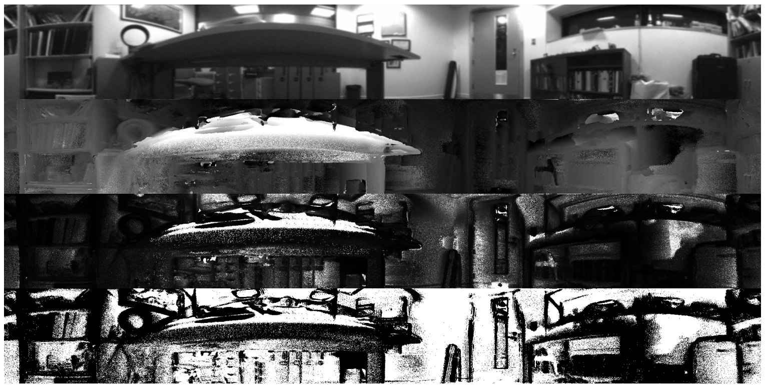
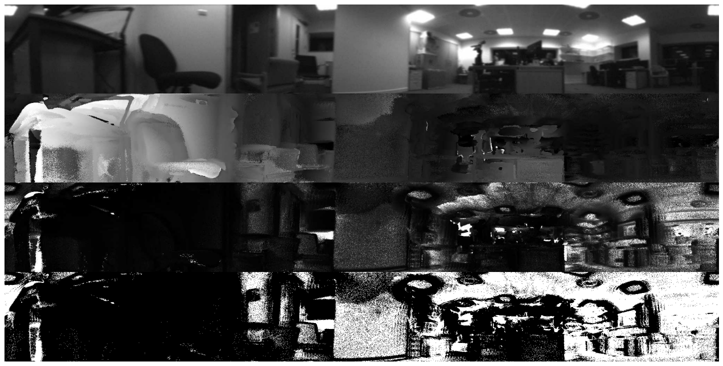
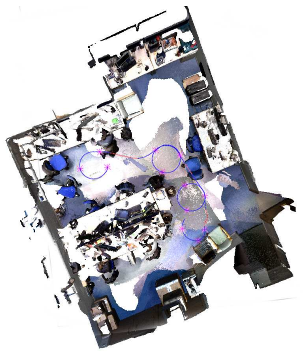


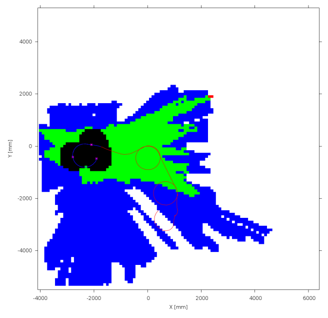
Fig. 4: Dataset “Cluttered Office”, from top to bottom: reference frame, estimated depth Map, estimated depth standard deviation map, depth threshold mask.



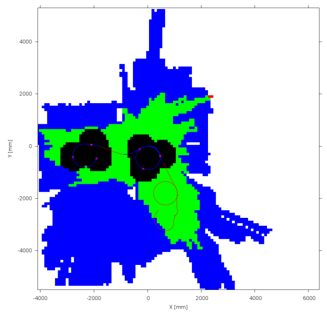
(a) Dataset “Large Office”, one of the measurements, from top to bottom: reference frame, estimated depth map, estimated depth standard deviation map, depth threshold mask.



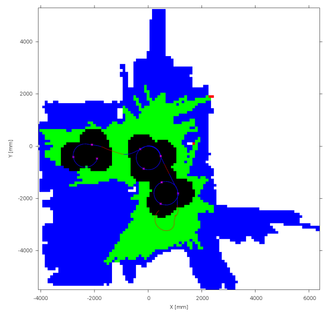
(b) Dataset “Large Office”: top-down view of the room for comparison (screenshot from an RGBD 3D SLAM system).



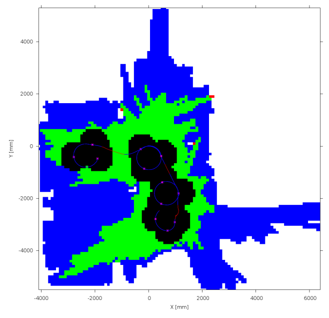
(c) Dataset “Large Office”: Occupancy grid, step 1



(d) Dataset “Large Office”: Occupancy grid, steps 1-2



(e) Dataset “Large Office”: Occupancy grid, steps 1-3



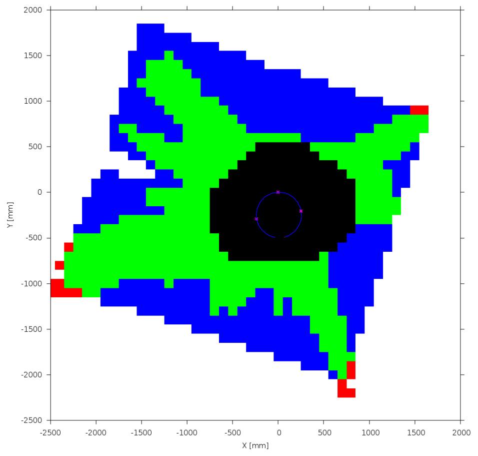
(f) Dataset “Large Office”: Occupancy grid, steps 1-4

Fig. 8

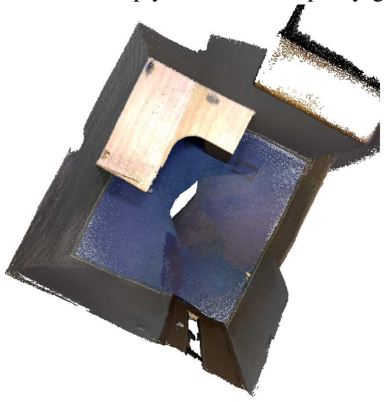
In this example it might appear that only 50% of the free space area was discovered (as compared to the laser), however here the laser scanner, due to its sensing characteristics and placement on the robot with respect to the camera, shines through chairs, under the desks and sometimes through the door-frame, heavily distorting actual drive-able free space area. Rather, we would like to highlight how such a large area of free space has been reliably found using a sensor which is not an obvious one for such a job, and in difficult real-world conditions. This gives great promise for getting more from the cameras already installed in many low-cost robots without the need for hardware changes.

V. CONCLUSION

We have presented a complete system for rapid mapping of free space around a mobile robot from a passive monocular omnidirectional camera. We illustrated the validity of the proposed approach with extensive results. The system was implemented in C++ with GPU acceleration, running on a laptop in near real-time, and we expect it to run on genuine low-cost embedded platforms in the near future. We rate the described method as promising for real-world applications,



(a) Dataset "Empty Office": Occupancy grid



(b) Dataset "Empty Office": top-down view of the room for comparison (screenshot from an RGBD 3D SLAM system).

Fig. 7

since it offers a viable alternative to expensive and inherently power consuming active cameras. As all passive stereo vision based systems, also ours will rely on sufficient texture in the scene. Furthermore, discoverable depth is limited by the chosen baseline – defined by the dimension of the circles that the robot follows.

Our system already estimates dense depth maps, depth standard deviation maps and a semi-dense point cloud. In future work, we thus would like to explore its use for more strongly modelled scene understanding and place recognition. Additionally, we would like to integrate active exploration capabilities so that the robot can autonomously and safely map an entire space.

ACKNOWLEDGEMENTS

Research presented in this paper has been supported by Dyson Technology Ltd.

REFERENCES

[1] G. Grisetti, C. Stachniss, and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *EEE Computer*, pp. 46–57, June 1989.

[3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[4] T. Whelan, J. B. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard, "Kintinuous: Spatially Extended KinectFusion," in *Workshop on RGB-D: Advanced Reasoning with Depth Cameras, in conjunction with Robotics: Science and Systems*, 2012.

[5] K. M. Wurm, D. Hennes, D. Holz, R. Rusu, C. Stachniss, K. Konolige, and W. Burgard, "Hierarchies of octrees for efficient 3D mapping," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2011.

[6] R. A. Newcombe, S. Lovegrove, and A. J. Davison, "DTAM: Dense Tracking and Mapping in Real-Time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[7] C. Geyer, "Catadioptric projective geometry: theory and applications," Ph.D. dissertation, University of Pennsylvania, 2003.

[8] J. P. d. A. Barreto, "General central projection systems: Modeling, calibration and visual servoing," Ph.D. dissertation, University of Coimbra, 2004.

[9] C. Mei, "Laser-augmented omnidirectional vision for 3D localisation and mapping," Ph.D. dissertation, INRIA Sophia-Antipolis, 2006.

[10] D. Scaramuzza, "Omnidirectional vision: from calibration to robot motion estimation," Ph.D. dissertation, ETH Zurich, 2008.

[11] L. Bagnato, P. Frossard, and P. Vandergheynst, "A variational framework for structure from motion in omnidirectional image sequences," *Journal of Mathematical Imaging and Vision*, vol. 41, no. 3, pp. 182–193, 2011.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[13] J. Engel, T. Schoeps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[16] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing (IVC)*, vol. 30, no. 2, pp. 65–77, 2012.

[17] S. Agarwal, M. K., and Others, "Ceres solver," http://ceres-solver.org.

[18] J. Zienkiewicz and A. J. Davison, "Extrinsics Autocalibration for Dense Planar Visual Odometry," *Journal of Field Robotics (JFR)*, 2014.

[19] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[20] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison, "Real-Time Camera Tracking: When is High Frame-Rate Best?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.